The SORCERER'S APPRENTICE

MISCELLANEOUS NOTES

Welcome to the fourth issue of this newsletter. I am happy to say that material has been pouring in every day, and that the future looks very bright. This is not to say that all is perfect however, and I hope you will excuse my lateness in getting around to some things, such as answering your mail. School and college life in general is taking up a good chunk of my time.

Ron Miranda (5812½ Pine Ave., Maywood, CA, 90270) is now forming a Sorcerer users group in the southern California area. The first meeting was held at the Hawthorne store of Computer Components of South Bay (15818 Hawthorne Blvd., Lawndale, CA) on Monday, September 24. For more information, get in touch with Ron at the above address or call him at (213)-771-0276.

Individuals may advertise in The Sorcerer's Apprentice for non comercial items (used equipment, etc.) for $1.00 per type-written line. Commercial rates are $30.00 for a full page, $18.00 for a half page, or $10.00 for a quarter page or fraction thereof.

Steve Long is still putting out his fine newsletter. The latest issue, his tenth, includes a 2 hex character search routine that can be used in conjunction with the monitor. Send Steve an SASE at 792 Laurie Ave., Santa Clara, CA, 95050 for subscription information.

Several people closer to the Michigan area have expressed an interest in having group meetings. Newman Computer Exchange here in Ann Arbor has offered the use of their store facilities on the second and fourth Thursdays of every month at 5:30 pm (they close at 7:00). If possible, I would like to arrange for meetings once a month, and I'd like to hear from interested people. Although no date is set as of yet, I would like to shoot for the 11th of October as a possible first meeting. Naturally, this will only take place if enough interested people let me know that they will be attending. If you can make it, let me know soon!

John Palevich (6200 Swords Way, Bethesda, MD, 20034) has had some success producing music on his Sorcerer:

"The obvious, cheap way to produce music on a Sorcerer is to wire an 8-ohm speaker between ground and any of the 8 bits of the parallel output port. This speaker will click whenever the bit it is connected to changes from a 0 to a 1 or from a 1 to a 0. For instance,

```
10 PRINT "SORCERERS CAN BUZZ."
20 OUT 255,0
30 OUT 255,255
40 GOTO 20
```

will produce a low pitched buzz. Any random hacker can expand this into a music program by some judicious machine language programming. Even I did it. Unfortunately, I have just moved to college and my computer will not catch up with me for several weeks.

"Anyway, there are two disadvantages to the click method of music synthesis. First, the quality of the music is laughable, and second, you can only synthesize one voice at a time (like playing the piano with only one finger).

"A far better technique is to connect an 8-bit D/A converter to the output port of the Sorcerer. I bought the Kl002 8-bit Audio Digital to Analog Converter from Micro Technology Unlimited (841 Galaxy Way, PO Box 4596, Manchester, NH, 03108) for about $40.00. One has to solder ten wires between a 22 pin connector and an RS-232 connector, total cost about $50.00. One "minor" catch is that you must write all the software yourself, but an article in the August 1977 Byte explains exactly how to do it. I have, using that article, the board, a Sorcerer, a small speaker and my native intelligence, concocted a beautiful program that simulates a 4 voice (4 different notes at once) Hammond-type electronic organ. (This really beats the Apple demo programs!) When I use the graphics to produce staff displays, I'll really have something."

Tom Bassett (253 Franconian Dr. W., Frankenmuth, MI, 48734) has made some excellent modifications to the database management program that appeared in the last issue. Both versions will be available from the software exchange (more about that later). Tom's additions include a delete option, and a more advanced method of changing the inputs and outputs that does not resort to any machine language routines. Tom also has passed along an amortization program and a program to aid the monetarily incompetent (such as myself) in balancing their checkbooks. Both of these latter two are rather long and will not be published, but will be available from the software exchange.

Software is finally appearing for the Sorcerer. I will be sending letters to all of them about advertising and the posibilities of doing reviews. Here are the ones I know of at the present:

CHB Software (1026 W. 26th St., San Pedro, CA, 90731) is producing some programs under license from Creative Computing: Slot Machine goes for $9.95; the Super Star Trek form 101 Basic Games is available for $14.95. Both are available for the package price of $21.50

Adventure International (Box 3435, Dept. X, Longwood, FL, 32750) is producing their popular Adventure game for the Sorcerer for $14.95. A friend has a copy of the tape, but has had loading problems. The folks at A.I. have been very cooperative and are sending a copy at 300 baud. I have played Adventure on some of the larger computers here on campus, and can easily say that it is one of the best computer games I've come across.

Northamerican Software (Box 1173, Stn. 'B', Downsview, Ontario, Canada, M3H 5V6) has 3 programs for sale: Stranded in Deep Space ($14.95), Programmable Graphic Scratch Pad ($9.95), and Know Your Sorcerer (part 1-8K, $7.95). They are available separately or for the package price of $29.95. Send a certified check or money order only.

PROGRAM EXCHANGE GETS UNDERWAY

Finally, no more typing and typing and debugging, etc., of lines of code to play your favorite game. Software listed in this newsletter is available on tape from Ralph G. Ruh, 623 Medill Ave., Lancaster, OH, 43130. Programs (cassettes only) can be exchanged on a 1 for 1 basis, or for the price of $2.00 for the first program and $1.00 for each additional program on the same tape. A minimum of 4 and a maximum of 6 programs per tape needs to be enforced to offset the cost of the tape and postage. Programs will be recorded twice at each baud rate. Here is the currently available software:

CATALOG NUMBER	NAME	FILE NAME	DESCRIPTION
S001	AMLOG	AMLOG	AMATEUR RADIO LOG BOOK KEEPER
S002	ZAP	ZAP	ROBOTS GAME
S003	ACCOUNTS	ACNTS	RECORDS HOUSEHOLD ACCOUNTS
S004	BILLIARDS	BLRDS	BUMPER POOL GAME
S005	ALIEN	ALIEN	CAPTURE GAME
S006	WAMPUS	WAMPS	GAME
S007	RACETRACK	RACET	HORSE RACING GAME
S008	JOUST	JOUST	JOUSTING GAME
S009	LUNAR	LUNAR	LUNAR LANDING GAME

Please use the catalog numbers when ordering. Some programs that appeared in earlier issues of the Apprentice have not been listed here because I just got the tape of the off to Ralph, but they should be available by the time the next issue comes out. If you have programs to donate to the exchange send them directly to Ralph; if you have relatively short progams that you would like published send them to me and I will forward them to Ralph. Remember, this is your group, and the more you can put into it, the more you will be able to get out. In the future, an updated list of available programs will be printed each issue, similar to the above, but the author's name will also be included. Also, we have still not resolved the copyright dilema, so some programs are going to have to stay shelved until we figure out a solution.

RENUMBERING PROGRAMS From Devin Trussell, Australia
    "This program must be appended to the program which is being
renumbered (see the last issue for this trick). It is executed
by 'RUN 63000'. The program is based on "Line Renumbering on the
PET" which appeared in the March, 1979 issue of Personal Computing.
In addition, 3 enhancements have been made:
    -"THEN GOTO" statement sequences are correctly handled.
    - The increment form one line number to the next is user
      specified.
    - The beginning line number is user specified.
The latter enhancement enables mutually exclusive line numbers to
be given to two programs. This is a prerequisite for splicing
programs (see Program Merging in the August issue). There are
two main restrictions on the program:
    1 - only 256 line numbers can be handled. This can be altered
        by changing the dimension on OL in line 63010.
    2 - The line number increment from one line to the next must
        be less than 256.
    "An outline of the program follows. A modest familiarity with
hexadecimal numbers on the part of the reader is assumed.
    "OL is an array containing the olk line numbers. The nth element
is the nth line number in the original program. OM is the index on
array OL. The final value of OM is the total number of lines cor-
rected (see line 63510). NH and NL are the high and low order bytes
respectively of the new line numbers. Al and Ah are the low and
high order bytes respectively of the hexadecimal address of the be-
ginning ot the next line in the program. They are initialized as 213
and 1 (which is 0D15), AD (see line 63020) is the decimal value of
AH and AL. The second (LL) and third (LH) address following AD
contain the low and high order bytes of the current line number.
    "Lines 63000-63019 are program initialization. In lines 63020-
63070 the old line numbers of the program are changed to the
desired new values. By means of a table look up (see line 63800),
the internal references (ie numbers following THEN (ASCII 162),
GOTO (ASCII 137) and GOSUB (ASCII 141) statements) are corrected
in lines 63500 to 63890. The program will give a "NO ROOM" mes-
sage (see line 63830) if there is insufficient space to insert the
new internal line numbers. The solution is then to manually cor-
rect the lines so affected. In most cases this chore can be a-
voided if one or two spaces are inserted ahead of all internal
line numbers when the original program is composed. The program
will give a "COULDN'T FIND" message if an internal line number
reference in the original program cannot be matched with an ori-
ginal line number. Note that II is used to specify the line
number increment. It should not be confused with the number
eleven. II occurs in lines 63019, 63050, and 63820.

The

Sorcerer's

Apprentice

courtesy

Pete Amhrein

```
LIST

63000 REM RENUMBER PROGRAM
63010 DIM OL(255):OM=0:AL=213:AH=1
63015 INPUT "NEW LINE NUMBERS BEGIN AT";ZZ
63017 NH=INT(ZZ/256):NL=ZZ-NH*256
63019 INPUT "LINE NUMBER INCREMENT IS"; II
63020 AD=256*AH+AL: LL=PEEK(AD+2): LH=PEEK(AD+3): OL=256*LH+LL
63030 IF OL=63000 GOTO 63500
63040 OL(OM)= OL:OM=OM+1
63050 POKEAD+2,NL:POKEAD+3,NH:NL=NL+II:IFNL>255THENNL=NL-256:NH=NH+1
63060 IF OM>255 GOTO 63500
63070 AL=PEEK(AD):AH=PEEK(AD+1):GOTO  63020
63500 REM LINE NUMBERS ALTERED. INTERNAL REFERENCES NOW CHANGED
63510 PRINT OM;"LINE NUMBERS CORRECTED": OM=OM-1: L=468
63520 L=L+4:LN=256*PEEK(L)+PEEK(L-1):IF LN=63000 THEN PRINT"FINISHED":END
63530 L=L+1:CH=PEEK(L):IF CH=0 THEN 63520
63540 IF (CH<>137)AND(CH<>141)AND(CH<>162) THEN 63530
63550 LO=L
63560 L=L+1:CH=PEEK(L):IF(CH=137)OR(CH=141)THEN63550
63565 IF CH=32 THEN 63560
63570 IF (CH>47) AND (CH<58) THEN GOSUB 63700: GOTO 63560
63580 IF N$=""THEN 63530
63590 IF CH=44 THEN GOSUB 63800: GOTO 63550
63600 GOSUB 63800: IF CH=0 THEN 63520
63610 GOTO 63530
63700 N=CH-48: N$=N$+RIGHT$(STR$(N),1):RETURN
63800 J=-1: N=VAL(N$): FOR I=0 TO OM: IFOL(I)=NTHENJ=1:I=OM
63810 NEXTI:IFJ=-1THENPRINT"COULDN'T FIND LINE #";N;"IN TABLE":GOTO63890
63820 NL=II*J+ZZ:NL$=STR$(NL):NL$=RIGHT$(NL$,LEN(NL$)-1)
63830 IFLEN(NL$)>(L-LO-1)THENPRINT"NO ROOM TO REPLACE";N;"WITH";NL:GOTO63890
63840 IF LEN(NL$)<(L-LO-1) THEN NL$=NL$+" ":GOTO 63840
63850 FOR I=LO+1 TOL-1:N$=MID$(NL$,I-LO,1):N=VAL(N$)+48:POKEI,N
63860 IF N$=" " THEN POKEI,32
63870 NEXT I
63890 N$="": RETURN
READY
```

...DT
Devin Trussel
Australia

DEAR MOTHER ITS A BUG....

In our last issue, an article appeared which described how to
recover from the loss of a program due to accidentally entering
NEW or CLOAD.... well, a few lines got left out, not to mention
a listing...sigh...just like in the big magazines. However, if
you turn now to page 13 (what else?) of the last issue, you will
see, about three and a bit lines up from the end, the words"..are
Entered in 1D6."   Insert here the following text:-

"Having done this, it is necessary to restore the end-of-program
pointer, which is located at 1B7 and 1B8. The address to be
entered in these two bytes is that of the first byte following
the zeroes that appear in memory following the last line of the
program. The least significant byte goes into 1B7, and the most
significant into 1B8."

...Incidentally further details of this operation are to be found
in paragraph 6 on page 7 of the same issue, relating to linking
programs.

...IM
Ian McMillan, Australia

This is good stuff if you want to reproduce characters in large sizes, because it gives you the character formation patterns without having to re-generate them.

Every character is stored as a pattern of eight consequtive bytes in ROM or in the special user definable character RAM.

The expression for Y finds the address of the top line of the character(actually the address is -Y). The routine then looks at that, and the next seven bytes in memory, and prints them out. The bit pattern of these bytes form the shape of the character displayed.

```
10 INPUT"ENTER CHARACTER";A$
20 Y=2048-8*ASC(A$)
30 PRINT"THE TOP LINE ADDRESS IS ";-Y
40 FOR X=Y TO (Y-7) STEP -1
50 PRINT PEEK(-X)
60 NEXT
70 GOTO 10
```

To user-define characters from BASIC , select your keytop, use the above routine to find the addresses of the eight bytes of the GRAPHIC/SHIFT character you are going to define. Then POKE the decimal values of the bytes required to make your bit pattern, into each location in turn. In a program, a DATA line and a READ in a loop very like the one in the program above will do the trick.

The "chunky graphics" routine in this issue can be used to create large characters using whole character blocks instead of little dots.

                                         ...IM
                                 Ian McMillan
                                  Australia

CHUNKY GRAPHICS

The following subroutine(s) allow the equivalent of the TRS80 SET and RESET commands.

The horizontal axis allows X values from 1 to 64.
The vertical axis allows values from Y=1 to Y=30.
To SET...GOSUB 990
To RESET...GOSUB 1000

```
990 Z=177:GOTO 1010
1000 Z=32
1010 P=3969-X-((Y-1)*64)
1020 POKE-P,Z
1030 RETURN
```

In your program, establish the values of X andY and call the subroutine, which can be renumbered to suit yourself.

                                         ...IM
                                 Ian McMillan
                                  Australia   6

THE WORD PROCESSING CORNER

This column is an experiment. If the readers like it, I will keep it going--if not, you are reading the first and last production.

What I propose for this corner is to provide information about the Exidy Word Processor. "Well", you say, "I can get that by reading the manual." Possibly, but we have had the EXWP going in our law office since the first EPROM's came out in June. We have been working with it about 200 hours per month. As you can imagine, we have gotten good at it.

This column proposes to be a combination of a) explanations of the various functions of the system (including a few tricks you won't find in the manual; b) question answering for any of the readers who are having problems and c) swap of information.

The processor has programming capabilities and it is not overdoing it to say that a "club" of users who have written or want to write macro programs for the EXWP would be a useful thing. As an example, here is a dandy little tape verification routine:

Suppose you have a 32K Exidy and you are going to put together, from scratch, a document of about 25,000 characters, and you have no motor control for your cassette. You have been slaving over this opus since dawn and now it is getting late and you have to get it on tape. The trouble is, you've got a) a cassette that every once in a while puts a little garbage on your tape or b) a joker on the floor below you who sometimes turns on a static producing defective electric razor which you cannot hear when he does it. You cannot load and unload this much text in the holding buffer while you check the tape because it will empty itself when the memory overloads. About all there is left is to record it in two or three dumps and pray that one of them is okay.

Instead, try this: record the text in the usual way by typing "w/c2" on the COMMAND line and then CR. When the program asks for "#/FILE NAME", type "1/", followed by CR. (I am assuming that your tape unit is set up as unit 1.) When the tape is recorded and with the text still in the computer, bring it back like so: type "100 R5/D20" and then CR. Turn on the tape. The processor will read tape five lines at a time and then delete what it has read. (Remember, a read automatically comes in at the bottom of text so it will not affect what is already there.) If there is a glitch on the tape, you will get "DEVICE ERROR" before the read is complete so you know you have to write the tape again. If the tape is okay, there will be a complete read and then "END OF FILE" and a request for a file name. You ignore that and press the escape key which will return control of the text to you.

Setting Monitor Parameters from BASIC

The monitor parameters set the display speed, Baud rate, tape header information, and select the input and output ports that are to be used by the programs running on the machine.

The "selection of Input or Output port" is somewhat misleading because what actually happens is that the routine in monitor that accesses the port in question is called. When O or I are set to an address (XXxx) it means that you have to provide, at that address, a machine language routine that will look after whatever device you have in mind.

In order to set the parameters from BASIC, one or two bytes have to be POKEd into the appropriate locations in monitor RAM, which is located at the top of memory. The actual addresses depend on the amount of memory in the system, so that the following tabulates in terms of 8, 16 and 32 Kilobytes.

1. SEt S=XX..to set video display rate

	8K	16K	32K
POKE 0 to 255 into........................	8143	16335	32719

Note: 0 is fast, 255 is slow.

2. SEt T=X...set Baud rate.

	8K	16K	32K
POKE 64 for 1200 Baud, POKE 0 for 300...	8142	16334	32718

3. SEt FF=XX...to set file type(in tape header) to XX

	8K	16K	32K
POKE 0 to 255 into.......................	8158	16350	32734

Note: BASIC will only load file types B0 to BF
      File type D8 will not auto execute.

4. SEt X=YYyy...set auto-execute address in tape header

POKE decimal equivalent of desired program starting address bytes, YY in H , yy in L

	8K		16K		32K	
	8163/4		16355/6		32739/40	
	L	H	L	H	L	H

Note: Some uncertainty with this one.

5. Selecting Output Port

Pairs of bytes are POKED into two addresses:-
8K...8144,8145   16K...16336,16337   32K...32720,32721

The bytes to be poked, in the same order as the addresses, above are as follows:-
SEt O=V (video) 27 and 224
    O=P (parlot) 33 and 224 (parralel output)
    O=L (ce.dvr) 147 and 233 (Centronics Printer)
    O=S (outape) 18 and 224 (tape output)
    O=XXxx   xx and XX (address of special output routine)

Note: XX is the decimal equivalent of the most significant byte, and xx is that of the least significant byte.

It is a lot cheaper than a disk system.

Here is another piece of information: you can get a set of replacement keys for the touch pad which have the editing commands engraved on them. Very handy if you are not the kind who remembers where they all are. A complete set, including the arrows, can be obtained from ARKAY Engravers, Inc., 2073 Newbridge Road, Bellmore, New York 11710 for $9.50 or $6.50 without the arrows, both post paid. The keys are engraved with the standard editing commands for the word processor and come in red, black, beige, blue or gray. This is also a good opportunity for someone who wants to get some special keys made up for the main keyboard.

If you want this fledgling column to continue write to David Bristor and maybe it will be back next month.

STEVEN GURALNICK
Attorney at Law
15 Southgate Avenue, Suite 246
Daly City, California 94015
Telephone: (415) 992-9200

6. Selecting Input Port

Pairs of bytes are POKEd into two addresses:-

    8K...8146,8147   16K...16338,16339   32K...32722,32723

The bytes to be poked, in the same order as the addresses are:

```
Set I=K (keybrd)  28 and 235
    I=P (parlin)  118 and 231
    I=S (intape)  218 and 226
    I=XXxx            xx and XX
```

Note: I repeat the point that these are the addresses of routines, not ports.

I hope that this information will be useful, but must point out that much of it is untried, and has been derived from study of monitor dis-assemblies, and may contain errors of detail, or calculation. I would be greatly obliged if any such errors were broght to my attention.

Example of usage

```
20 POKE32719,100
30 REM SLOWS PRINTING SPEED ON SCREEN
40 POKE16336,147:POKE16337,233
50 REM MAKES THE CURRENT OUTPUT PORT CENTRONICS
```

The Technic Guide to
Assembler Language

courtesy The Michigan Technic , Jeff Diewald, and "Johnson"

Many computer engineers have discovered a definite lack of computational power in most modern machines. Often a simple command could eliminate large, bulky routines. A number of fellow engineers have proposed the following list "additional" commands that (we feel) would tremendously expand the scope and power of the typical computer. The instructions are given with easy-to-remember mnemonics for quick addition to a macrolibrary or instruction set.

AAC Alter All Commands
AAR Alter At Random
AB Add Backwards
AFVC Add Finagle's Variable Constant
AIB Attack Innocent Bystander

AWTT Assemble with Tinker Toys
BAC Branch to Alpha Centauri
BAF Blow All Fuses
BCIL Branch Creating Infinite Loop
BDC Break Down and Cry

BF Belch Five
BDT Burn Data Tree
BW Branch on Whim
CBNC Close, But No Cigar
CH Create Havoc

CMD Compare Meaningless Data
CML Compute Meaning of Life
CNB Cause Nervous Breakdown
COLB Crash for Operator's Lunch Break
CRASH Continue Running After Stop or Halt

CS Crash System
CSL Curse and Swear Loudly
CVG Convert to Garbage
DBZ Divide by Zero
DDC Dally During Calculations

DMPE Decide to Major in Phys. Ed.
DOC Drive Operator Crazy
DLN Don't Look Now . . . . .
DPMI Declare Programmer Mentally Incompetent
DPR Destroy Program

DTC Destroy This Command
DTVFL Destroy Third Variable From Left
DW Destroy World
ECO Electrocute Computer Operator
EIAO Execute in Any Order

ENF Emit Noxious Fumes
EP Execute Programmer

FLI Flash Lights Impressively
FSM Fold, Spindle, and Mutilate
GCAR Get Correct Answer Regardless

GDP Grin Defiantly at Programmer
HCF Halt and Catch Fire
HCP Hide Central Processor
ISC Insert Sarcastic Comments
JTZ Jump to Twilight Zone

LAP Laugh at Program(mer)
LPA Lead Programmer Astray
MAZ Multiply Answer by Zero
MW Malfunction Whenever
MWT Malfunction Without Telling

OML Obey Murphey's Laws
PEHC Punch Extra Holes in Cards
PNRP Print Nasty Replies to Programmer
RA Randomize Answer
RCB Read Commands Backwards

RDA Refuse to Disclose Answer
RLI Rotate Left Indefinitely
RPM Read Programmer's Mind
RRSGWSSNK Round and Round She Goes,
            Where She Stops, Nobody Knows
SAI Skip All Instructions

SCCA Short Circuit on Correct Answer
SFH Set Flags to Half-mast
SFT Stall for Time
SOS Sign Off, Stupid
SRDR Shift Right, Double Ridiculous

TARC Take Arithmetic Review Course
TLO Turn Indicator Lights Off
TN Take a Nap
TPDH Tell Programmer to Do it Him/Herself
TTA Try, Try Again

UP Understand Program
WSWW Work in Strange and Wondrous Ways

```
10 CLEAR 2000
20 REM *** ALLOW A GENEROUS AMOUNT OF STRING SPACE
30 REM *** PROGRAM BY TONY AUGUST
40 PRINT TAB(14)"A SIMPLE PATTERN GENERATION PROGRAM"
50 PRINT: PRINT: PRINT
60 PRINT TAB(10)"THIS  PROGAM WILL GENERATE MANY INTERESTING"
70 PRINT
80 PRINT TAB(10)"AND  UNSUAL PATTERNS.  THE USER MAY ENTER 5"
90 PRINT
100 PRINT TAB(10)"LINES OF 64 CHARACTERS EACH MIXED WITH "
110 PRINT
120 PRINT TAB(10)"LETTERS, NUMBERS, EXIDY SORCERER GRAPHICS,"
130 PRINT
140 PRINT TAB(10)"OR USER DEFINED GRAPHICS.  ANY THING"
150 PRINT
160 PRINT TAB(10)"WHICH CAN BE PLACED ON THE SCREEN WILL BE"
170 PRINT
180 PRINT TAB(10)"INTERWOVEN INTO AN UNSUAL VIEDO DISPLAY."
190 PRINT
200 PRINT TAB(10)"TO 'REPLAY' THE PREVIOUSLY GENERATED PATTERN"
210 PRINT
220 PRINT TAB(10)"PRESS THE RETURN KEY."
250 PRINT
500 INPUT A$, B$, C$, D$, E$
510 IF A$="    " THEN 700
520 IF B$="    " THEN 700
530 IF C$="    " THEN 700
540 IF D$="    " THEN 700
550 IF E$="    " THEN 700
560 REM *** SPACE BETWEEN QUOTES IN ABOVE IS NOT IMPORTANT
700 FOR X=1 to 64
710 PRINT A$"    ";
720 PRINT B$"    ";
730 PRINT C$"    ";
740 PRINT D$"    ";
750 PRINT E$"    ";
760 REM *** MAKE CERTIAN THAT THE SEMICOLON IS IN THE ABOVE 5 LINES
770 NEXT
780 PRINT: PRINT: PRINT
790 PRINT "ENTER YOUR NEXT 'SEED' PATTERN"
800 PRINT
810 GOTO 500
```

Tony August, 5953 E. State St., Sharon, PA, 16146

Ralph G. Ruh, 623 Medill Ave., Lancaster, OH, 43130
(WD8CVO)

```
1 PRINT CHR$(12)
2 GOSUB 300
3 PRINT TAB(10);"AMATEUR RADIO LOG-AMLOG"
4 PRINT TAB(10);"    BY RALPH G. RUH"
5 PRINT TAB(10);"    LANCASTER, OHIO"
7 DIM A$(150),N$(150),B$(150),C$(150),D$(150)
8 W$="AMATEUR CALL: "
9 F$="NAME: "
10 R$="QTH: "
11 E$="DATE/TIME: "
12 G$="COMMENTS: "
14 GOSUB 300
15 GOSUB 300
16 PRINT:PRINT
17 INPUT"DO YOU WISH TO ENTER ANY NEW DATA?(Y OR N) ";Q$:PRINT
18 IF Q$="N" THEN 50
19 PRINT"ENTER YOUR DATA IN THE FOLLOWING FORMAT:":PRINT:PRINT
20 PRINT "NNNN DATA 'AMATEUR CALL',NAME,QTH,DATE/TIME,COMMENTS"
21 PRINT:PRINT
23 FOR X=1 TO 150
25 READ A$(X),N$(X),B$(X),C$(X),D$(X)
26 IF A$(X)="ENDATA" THEN 29
27 NEXT X
28 REM X=0
29 PRINT "THE NEXT DATA LINE TO BE ENTERED IS NUMBER";999+X;","
30 PRINT:PRINT
35 PRINT "NOW ENTER DATA, RESTART PROGRAM WITH RUN AND CR"
36 PRINT "WHEN FINISHED."
38 PRINT:PRINT:STOP
50 PRINT:PRINT "ANSWER WITH 'END' IF THROUGH.":PRINT
51 PRINT "ENTER AMATEUR CALL YOU ARE SEARCHING":PRINT
60 GOSUB 300
90 INPUT M$
100 IF M$= "END" GOTO 9999
110 FOR I = 1 TO 500
115 IF A$="ENDATA" GOTO 200
120 READ A$,N$,B$,C$,D$:P$=RIGHT$(A$,LEN(M$))
130 IF A$="ENDATA" GOTO 200
140 IF M$<>P$ GOTO 200
145 PRINT:PRINT W$;TAB(15);A$
150 X=LEN(N$):PRINT:PRINT F$;TAB(15);N$
160 IF N$="*" GOTO 200
170 X=LEN(B$):PRINT R$;TAB(15);B$
171 IF B$="*" GOTO 200
180 X=LEN(C$):PRINT E$;TAB(15);C$
181 IF C$="*" GOTO 200
190 X=LEN(D$):PRINT G$;TAB(15);D$
191 IF D$="*" GOTO 200
200 NEXT I
210 GOSUB 300
215 RESTORE:A$="1000"
220 GOTO 50
230 GOTO 9999
300 FOR I= 1 TO 63:PRINT"-";:NEXT I:PRINT:RETURN
1000 DATAWD8CVO,RALPH RUH,"LANCASTER, OHIO",081379 1715,HELLO DA
VE
8000 DATA ENDATA,*,*,*,*
9999 PRINT:PRINT "DON'T FORGET TO SAVE DATA ON TAPE":PRINT:END
READY
```

12